Section 3, Remarks:

REMARKS

Re-examination and reconsideration of this case is respectfully requested for the 20 claims now in this case in view of the amendments to the claims presented herewith, and the following remarks.

Applicant presents 20 claims herewith, the original claims 1 – 12, some of which have been amended, and new claims 13 – 20. All of the new claims are dependent from main claims, either the main system claim 1, or the main method claim 11.

Thus the system claims are claims 1 – 10 and 13 – 17, and the method claims are claims 11, 12 and 18 – 20. There are 15 system claims and 5 method claims.

Claims 1 - 6, 11 and 12 have been amended. No new matter has been introduced by the amendments or in the new claims, as discussed in detail below. The new claims are submitted to round out the suite of claims for more comprehensive coverage of the inventive subject matter. Entry of the amendments and the new claims is requested, and allowance of all claims is respectfully urged.

The conditional allowance of dependent claims 2 and 12 is gratefully acknowledged. However, writing these claims in independent form is deferred pending the Examiner's review of the amended main claims 1 and 11. It is thought that upon review, these claims will be deemed allowable, and therefore the rewrite of claims 2 and 12 will be deemed moot.

Response to Objection and the §112 Rejection:

Claims 2 and 12 were objected to because of presence of the phrase "(via get info)". In addition these claims were rejected under §112 as indefinite because of the inclusion of the phrase "in an inverse manner". This latter rejection overlooked claim 5, which also included the "inverse manner" language.

In response, both of those phrases were found to be redundant, and have been deleted where found in claims 2, 5 and 12. Accordingly, both the rejection and the objection have been rendered moot, and their withdrawal is respectfully requested.

Additional Claim Amendments:

With respect to the amendments to claims 1 – 6, 11 and 12, the comments below relating to claim 1 also relate to claim 11, and relating to claim 2 also relate to claim 12, it

being understood that the claims are appropriately directed to system apparatus and method of management of information, respectively.

In sub-part or step a) of claims 1 and 11, respectively, the redundant and confusing language "object store object-oriented database" was amended to the more direct, simpler and more clear form "database for storing objects". In addition, the "at least one" modifier in line 3 of part/step a) was inadvertently inserted at the end of the clause, and has been moved to the beginning of the clause listing types of objects and metadata. Since there are grammatical in nature, no new matter has been introduced.

Claim 1, part b) was amended to be consistent as to the antecedent for the database. The same is the case for Claim 1, part d), lines 3 and 4. In line 4, the term "link" grammatically belongs before the term "metadata". See the definition of link metadata in the Specification at page 5, lines 8 – 10. The term "different domain" has been inserted before "sources" in both part d) of claim 1 and step b) of claim 11 to make it clear the MFS system is able to manage information objects from very different domains. See the Specification at page 4, 5 and 41, for example. Actually, that is one of the key capabilities of the system and method.

In method claim 11, step b), the first three lines have been grammatically redrafted to more clearly identify the results of the object scanning and processing.

Part e) of claim 1 and the mirror part c) of method claim 11 have been added. Sub-part/ step i) has been imported from original claims 2 i) and 12 i), respectively; sub-part/step ii) has been imported from original claims 2 iv) and 12 iv), respectively; sub-part/step iii) has been imported from original claims 2 vi) and 12 vi), respectively. In addition, those sub-parts/steps of original claims 2 and 12, namely sub-parts i), iv) and vi) have been deleted from claims 2 and 12, and the other, original, remaining sub-parts of those claims renumbered. Accordingly, no new matter has been introduced by those amendments to the parts e) of claim 1 and step c) of claim 12.

Claims 2 and 12, part i) are amended to delete the "inverse manner language", and to clarify, that user categorizations are applied to objects in at least one container and by gathering cross-referenced sets of containers; see page 20, line 14 ff, and generally throughout the Specification for support. By this clarification it is clear that the inverse manner language is redundant, and accordingly was deleted. In part iv) the correct term is "working sets", see Specification, page 6, lines 16 – 19. In part viii) the language "notification to the user" was

clarified to recite that a notify event is provided; see the Specification at page 31, lines 29 – 31 and the discussion in connection with Fig. 35 for disclosure and discussion of exemplary notify events. In part ix) the link creation list has been supplemented by the recitation of user entry of names, metadata queries, clicking on collections to choose them and by the system matching metadata criteria. See, for example, support at page 10, line 34 through page 11, line 9.

Minor grammatical corrections have been made to claims 3 – 6 to correct the antecedent basis for the reference to the database of claim 1. In addition, the inverse manner language was removed from line 8 of claim 5. In claim 6, since the reference to database was made consistent with claim 1, and the database includes other than link metadata (annotation and derived metadata, for example), the term "only" was incorrect and so was deleted.

Claims 7 – 10 are original, un-amended dependent claims.

The amendments to claims 11 and 12 have been discussed above.

Dependent claim 13 is new, and is directed to details of the function of the **pInclusion** and **pExclusion** features that permit the user to include or exclude objects from a collection despite either not matching or matching criteria. This is supported, inter alia, in the exemplary discussion at page 30, lines 27 through page 31, line 7. See also the general discussion of **Classification** at page 32, lines 8 – 27.

Dependent claim 14 is new, and is directed to details of the function of the modification of properties of the object or containers. This is supported, inter alia, in the discussion of the **pContainers** updating function at page 31, lines 8 – 34, particularly the discussion of points 5 and 6 as examples.

Dependent claim 15 is new, and is directed to creating metadata representing dependent properties. This is supported, inter alia, in the discussion at page 35, lines 5 – 8 and page 36, line 31 through page 37, line 1, and the discussion of dependent properties 3508 in Fig. 35.

Dependent claim 16 is new, and is directed to maintaining consistency of collections as discussed throughout the Specification, an example being at page 11, line 33 through page 12, line 1; page 27, lines 13 – 18; and in connection with Fig. 32.

Dependent claim 17 is new, and is directed to the system running a process based on the event of a collection object set being changed. See, for example, the process threads initiated in ***consistency maintenance: updater, notifier, classifier*** and ***synchronizer***, as disclosed at page 27, lines 13 – 18.

Dependent method claims 18 – 20 are directed to rounding out the scope of the method claim 1, and mirror original claim 8, and new claims 13 and 14. No new matter is involved in the mirroring as method the original system claim 8. The support for new claims 13 and 14 has been discussed above, and is incorporated here by reference.

The new claims, being all dependent and directed to features clearly encompassed by the main claims 1 and 11, respectively and disclosed in the Specification and Drawings as filed, are directed to the same invention. The amendments help to more clearly and distinctly claim inventive features of the claimed system and method.

Accordingly, no new matter has been added to the amended and new claims.

Response to the § 103 Rejections of Claims 1 and 3 – 11:

It is Applicant's view that the rejections of all but claims 2 and 12 as obvious variations of a combination of 3 patents (Watkins, McCotter and Lewak), and as to claim 8, further in view of a 4th patent to Rochford, are inappropriate and should be withdrawn.

Applicant appreciates the detailed analysis set forth in the Office Action. However, as will be seen from the discussion below, critical elements and functionality of the references have been minimized, not fully understood in context, or omitted. Applicant endeavors to show below in a discussion of exemplary distinguishing features of his invention that the inventive MFS system and method are not taught or suggested by the references, taken alone or in combination.

Initially, it should be noted that the Office Action acknowledges that the collection based key-phrase hyperlinking and maintaining dynamic collections features of original claims 2 and 12, sub-parts i) and iv), respectively, are not shown in the art. These features have been imported into main claims 1 and 11, respectively. Accordingly, those claims should now be allowable.

In addition, the feature vi) of original claims 2 and 12 was analyzed in the Office Action as shown by Lewak. Applicant shows below that Lewak does not show or teach this feature. Accordingly, all three features of part e) of main claim 1 and step c) of main claim 11 are not shown or taught by the art of record, and should be allowable. Favorable action of allowance of all claims is therefore in order and is respectfully urged.

Note further that dependent claims 2 and 12 retain the "not found in the art" features: iii) sticky path; iv) working sets; v) extensible domain mechanism; vi) extensible mechanism

for managing attributes from files of different formats; and x) setting of property values of objects by drag-and-drop. Accordingly these claims are patentable on their own merit.

Turning now to the references and in response to the Examiner's comments on pages 3 through line 3 of page 6 of the Detailed Action, the principal references are Watkins and Lewak, as McCotter is merely relied-on to show a B-Tree. While the principal references do maintain links to categories for each object, ***in significant and patentable contrast***, the inventive MFS system and method provide a mechanism whereby the object's categories (***Collections*** in the current invention) may be automatically specified by metadata criteria that the object may or may not match. Neither Watkins nor Lewak teach or suggest, taken alone or in combination (including in combination with McCotter) the features of automatic, persistent and dynamic updating of collections. In the inventive system and method, by specifying such criteria for each collection, the objects are automatically categorized, or placed within the appropriate collections, without additional interaction by the user. Further, once the criteria are specified for each collection, objects that are newly created or modified are removed from, or added to, collections based on their metadata which either satisfies the collection criteria or does not satisfy it. The following table will assist in understanding the patentable distinctions relating to the features of the automatic, persistent and dynamic collection updating that are claimed in main claims 1 and 11, not present in any of the references:

Invention	**Watkins**	**Lewak**
Criteria specified by each collection automatically collect objects that match the criteria; the collection is always main-tained up to date, and changes in the collection can trigger events or scripts. Collections are persistent and dynamic-ally updated as the source objects change, are added, or removed. This is claimed in main claims 1 and 11, parts/steps e) ii), iii), and c) ii), iii), respectively.	Objects are indexed by metadata and full text; queries on the index cause the appropriate objects to be returned at that time. There is no concept of persistent, dynamic collections.	Objects may be categorized by the user into multiple categories; clicking on a category (or mul-tiple categories) causes a query to occur that returns the objects that exist in that category (or categories). There is no concept of criteria for a category, nor is there a concept of dynamic, persistent updating.

Further, in the Detailed Action in connection with claim 5 (page 7 of the Detailed Action) it is alleged that the combination of references teach "to automatically classify...". That is not correct. Lewak, et al does not classify or categorize objects except by whether the object has been previously classified manually by the user, and that is not automatic classification. There is no concept in Lewak or Watkins of criteria attached to a category (or ***Collection***

in the current invention), and no automatic classification. Lewak does allow the creation of "hybrid folders" – "containers" that show the conjunction of a number of categories - but does not provide a feature that results in the specification of criteria to determine the categorization, into multiple collections, of an object based on its metadata. Again, the following table highlights this significant difference, again referencing claim 1, e) iii) and claim 11, c) iii):

Invention	Lewak
Collections of objects are defined both automatically, via collection criteria that match object metadata, and manually, by user specification (drag-and-drop, typing collection names, or clicking on a grid of collection objects to choose collec-tions to which the object should belong). Manual additions and removals are stored with the collection, separately from objects that may belong to the collection by virtue of match-ing the collection's criteria .	Categories of objects are defined manually, by user specification (clicking on appropriate category names to apply those categories to the chosen file). Hybrid Folders are defined by logical operations (AND, OR, NOT) between categories

Further in that discussion on page 7 of the Detailed Action, it is incorrectly asserted that the principal references teach "to apply a user's categorizations in an inverse manner...". Not so. Neither Lewak nor Watkins provides the capability of choosing a collection and then, instead of showing the collection's objects, showing the ***related*** collections, that is, the collections that share objects with the given collection (Viewing By Reference). For example, given the "Bills" collection, that shows all bills received when viewing normally, the inventive system and method Reference view will show that there are bills received Today (that is, bills in the Today collection), bills received for Utilities, bills received for Car Payments, and so on. While Lewak does provide a feature by which selecting the first category restricts the other categories that are relevant, in contrast the inventive Reference View not only shows the relevant collections, but the objects within that collection that satisfy the criteria of both the main collection and the chosen sub-collection. The following chart is instructive:

Invention	Watkins	Lewak
From a collection or other container, the "By Reference " view shows the list of additional containers that share at least some of the objects in the container. Then by selecting one or more of these addition-al containers, only those objects that exist in the main container ***and*** the selected cross-reference container(s) are then shown.	No mechanism for showing inverse relationships to categories, since categories are not persistent.	"Search Filter Definition" state allows user to select categories for searching, but when a category is selected, only those categories that share at least one entry with the selected category remain. This does not show the objects cross-referenced by other categories.

In the discussion on pages 7 and 8 of the Detailed Action with reference to Claim 6, Lewak is incorrectly characterized as teaching that "clicking on the icon representing an object in one collection window and dragging it into another collection window to establish a new link..." (Lewak Column 8, line 61 - 67, Column 11 line 3 - 8, and Column 15, line 22-36). Again, not so. In contrast to the claimed invention, Lewak **only** provides a mechanism by which **a single file may be categorized by clicking** on relevant categories, thereby selecting them. In patentably distinct contrast, the inventive claimed MFS system and method provides features and steps that permit categorizing multiple objects at once, in several ways:

- by dragging one or more objects from a folder or collection window to another collection window;
- by clicking on relevant collections in an Annotation view, which lists all of the collections available; and
- by selecting one or more objects and choosing the Collect... command, then naming existing or new collections in which the selected objects should be grouped.

The following table is helpful to highlight these patentable distinctions:

Invention	**Watkins**	**Lewak**
Objects are categorized into collections automatically, by criteria, and manually by dragging from collection or folder windows to another collection window; by clicking on relevant collections in an Annotation view; and via the Collect command, which allows the user to name multiple destination collections at once.	Search by query only; no persistent categories.	Objects are categorized manually by clicking on category names; there is no mechanism of visual containers (e.g. folder or collection windows) for drag-and-drop categorization.

In connection with the discussion on pages 8 and 9 of the Detailed action in connection with claim 7 the references are mischaracterized as teaching the system of claim 3 to "process-[es] so as to query said metadata, including queries selected from matching key phrases in an object's text, matching dates and time ranges, or exact matches...". Again, not so. Lewak does not provide arbitrary metadata queries. His hybrid folders are only specified by logical set operations on existing categories. In the example given, (Lewak, Column 10, Line 19 through Column 11, line 2), the objects that result in the hybrid folder are the objects that exist both in the Memos category and the Jones category. In contrast, the current invention provides for criteria, associated with each collection, that automatically classify objects into the appropriate collection. The hybrid folder example, where the criteria could be specified as "category =

Memos AND category = Jones", is simply one of many possibilities for collection criteria in the current invention. Unlike Lewak, in the inventive MFS system and method, other criteria, such as "name contains 'jones' or category = 'urgent' and modified = today" could be used in the current invention to specify a collection. The chart below is helpful to show this patentable distinction:

Invention	Lewak
Processes new and changed objects and automatic-ally collects them into collections based on arbi-trary collection criteria. Automatically removes objects that have changed from collections where the objects' metadata no longer satisfy the collections' crite ria.	Only manual categorization is provided; there is no concept of criteria for categorization. Hybrid folders are a way to specify the logical set operation of categories of objects; however, they are not maintained up to date at all times, but simply provide a query to run when the user desires to see the Hybrid Folder's conte nts.

Further, in the Detailed Action discussion on claim 7 it is incorrectly argued that the references teach "dynamic updating of all relevant collections...so that any changes...result in timely and appropriate changes to affected object views and for hypertext generation...". While Lewak and Watkins each provide "watchdog" processes that note changes of objects on disk, neither describes automatic, dynamic updating of collections as the underlying data space changes. In the inventive MFS system, as objects change they may move from collection to collection (their categories automatically changed). In both Lewak and Watkins, any categorization is done by the user directly; selecting a category causes a query to run at that time to go to the database and fetch the appropriate objects. In the claimed MFS invention, the objects are stored persistently in collections, and are added and removed dynamically—this allows for features such as notification when an object changes its classification, or is added or removed from a specific collection; See claims 1 and 11, parts/steps e) ii), and c) ii), respectively

Invention	Watkins	Lewak
Collections are maintained up to date at all times, by dynamically checking new, changed, and removed objects against all collection criteria. The user may be notified when a collection changes its contents—when an object is added or removed.	Objects returned to view by specific query only. There is no feature of a persistent categorization or collection.	Objects returned by specifying a given category or categories in which the objects exist, thus creating a limited query based on category only. There is no feature of a persistent category container.

In connection with the rejection in the Detailed Action of claim 8 on pages 9 and 10, which adds the Rochford reference to the other 3 references, while Rochford shows a useful

program in which a containment hierarchy may be displayed, and remain visible while sub-items in the branch are scrolled up and down, the characterization of it teaching the claimed MFS system is not correct. The inventive claimed MFS system has the following patentable differences and advantages:

- in Rochford, only the contents in the currently selected branch are visible; in contrast in the claimed MFS system the contents in the branch are visible, as are all items that are siblings and parents to the branch that are sorted lower in the display;
- In Rochford, you can only scroll within the current branch's contents; in the claimed MFS system you can scroll within the entire tree using the exact same scrolling mechanism that traditional outline displays use;
- The claimed MFS system provides a much simpler interface - **a sticky region** at the top of a traditional, scrolling outline view - and provides **a dynamically updating view** of the currently-visible branch as the user scrolls;
- The claimed MFS system allows the user to determine whether specific branches should be traversed or not during scrolling, by providing a standard mechanism for disclosing selected branches (opening folders or containers in the outline) and not disclosing others. Non-disclosed branches do not stick, and their contents are not traversed. The chart below assists in understanding the patentable distinctions:

Invention	**Rochford**
Extends the outline view with a "**sticky path**" that remains inside the outline frame; the portion of the outline below the path continues to scroll normally. The **sticky path area changes dynamically** during scrolling to indicate the hierarchy above the currently-visible items immediately below the sticky path. Items of all levels are visible below the sticky path, providing context. Clicking on a sticky path item causes the outline to automatically scroll to the beginning of the selected sub-path.	Provides an **external** window or pane that displays the hierarchy above items that are shown in an independent pane. **Moving up in the hierarchy must be done by clicking in the hierarchy pane, rather than simply scrolling**. Scrolling is only permitted within the currently-selected branch of the hierarchy.

Further, with respect to Lewak, as applied in all the rejections, it is significant to note his teaching (Lewak, column 8, lines 6-15) that the user is required to categorize a file, including a new import file, if it hasn't been already categorized (by the user previously). This is a clear teaching to go in a direction 180° opposite Applicant's MFS system. That key teaching of Lewak is clearly NOT automatic generation of collections by one or more object content attributes. In the claimed MFS system, when a new object is observed to exist, it is automatically placed in the appropriate collections (categories) based on the criteria for each

collection. In addition, in the inventive MFS system, the user may manually categorize objects by dragging them into collections or otherwise specifying which collections should contain the given objects; this independent user categorization, maintained separately from the automatic categorization by collection criteria is a dual feature not shown in the references.

In Lewak, categories are organized alphabetically and under column headings (Lewak, column 8, lines 16 – 30). That is distinct from the claimed MFS system in that the invention provides real time filtering/sorting (see claims 2 vii) and 12 vii) having features and steps to:

- select for display only those objects that match a given filter specification, updated dynamically while the user types;
- show only those objects that satisfy both the criteria of the current collection and those of other containers that both contain the given objects, updated dynamically as the user chooses collections within the view;
- sort by any ***sortable*** property of the currently visible objects (e.g. name, kind, creation date, count of sub-objects, and so on). While many systems allow sorting on a small number of basic file system properties (name, dates, and so on), the current invention extends this ability to *any* sortable property. The current invention maintains a ***sortable*** attribute of property specifications, and ***comparable*** operations in order to enable sorting of arbitrary attributes.

With respect to claims 2 and 12, part/step viii) recitation of providing a notify event of collection establishment and changes in collections", Lewak's reference to "notify the user" in his Column 7, lines 49 - 67 and Column 8, lines 6 – 15 means that the user is simply notified when an object ***requires*** categorization (e.g., when it is created or modified in the Lewak file system). Since Lewak does not have automatic categorization of objects, the only changes in collections that might occur are those re-categorizations done manually by the user. In contrast, the inventive claimed MFS system, by virtue of automatic, dynamic categorization of objects, can re-categorize objects without user intervention, and then notify the user of the results of that re-categorization upon or after that re-categorization event has occurred. That is not possible in Lewak.

Accordingly, it is Applicant's view that the references do not teach or suggest the claimed MFS system or method, and accordingly the rejections should be withdrawn and all claims allowed.

Applicable Principles of Law for Combination of References Rejections:

The fundamental principle, as articulated by the Court of Appeals for the Federal Circuit in **In re Gordon,** 221 USPQ 1125 (Fed. Cir. 1984), is that the prior art **must** suggest the combination of references. In **Gordon,** the Court rejected the idea that the prior art devices **could be** modified to produce the claimed device as a proper basis for an obviousness rejection, holding **the combination is not proper unless the prior art suggests the desirability of such a modification**. In **SmithKline Diagnostics, Inc. v. Helena Laboratories Corp.**, 8 USPQ2d 1468 (Fed. Cir. 1988), the Court held that to pick and chose elements from references to recreate the invention is **not** proper. And in **Northern Telecom, Inc. v. Datapoint Corp.**, 15 USPQ2d 1531 (Fed. Cir. 1990), **cert. denied,** 498 U.S. 920 (1990), the Court held that "[i]t is insufficient that the prior art disclosed the components of the patented device, either separately or used in other combinations; there must be some teaching, suggestion, or incentive to make the combination made by the inventor." (Emphasis added).

These governing principles were applied by the Court in holding in error the obviousness rejections in **In re Bond**, 15 USPQ2d 1566 (Fed. Cir. 1990) and **In re Mills**, 16 USPQ2d 1430 (Fed. Cir. 1990). **In re Mills** specifically held that although the prior art device could be modified to run the way the applicant's device was claimed to run, "there must be a suggestion or motivation in the reference to do so." 16 USPQ2d 1430. Since there was none, the rejection was in error and was reversed. More recently, in **Sensonics, Inc. v. Aerosonic Corp.**, 38 USPQ2d 1551 (Fed. Cir. 1996), the Court reiterated this principle, holding there was no teaching or suggestion in the prior art that would have led a person skilled in the art to select the specific mechanical and electrical structures and concepts and combine them in the manner of the invention of that case.

As a further principle, both the Courts and the Board of Appeal have long held that the suggestion for the combination in the references cannot come from the Applicant's Specification, see, for example, **Ex parte Brack**, 134 USPQ 445 (POBA 1961). The reason is simple: Applicant's Specification is not prior art. **Applicant's specification cannot be used as a parts-list to search for disparate parts in the art, and then used as a blueprint to assemble the selected parts.** The sources for the motive to select the parts and to reassemble them to obtain the desired result must come from the references.

The above principles were not followed in this Office Action. There is no teaching in

either reference pointing to the other. While the Office Action quotes Rochford, Col 2, lines 8 – 11 in regard to navigating "through file folder contents in a manner which allows the contexst of what is being observed to be continuously clear", the manners of Rochford, and of Watkins, McCotter and Lewak is not the claimed manner of Applicant. Combining the references does not product the claimed MFS system. The result is that the rejections are unsound and should be withdrawn.

The remaining references cited but not applied: Eliacott, Hobbs and Bowen et al., taken alone or in combination with the principal references of Watkins and Lewak, and/or McCotter and Rochford do not show or teach the claimed inventions.

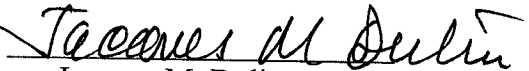
CONCLUSION

It is Applicant's position that the case is now in complete condition for allowance as the rejections are unsound or have been rendered moot. To assist the Examiner in evaluating the art cited and to point out the distinguishing functionality in the amended claims, Applicants have requested the courtesy of a Telephone Interview upon the Examiner reaching this case.

Favorable action of allowance is respectfully requested. In the event that there remain any open issues, the Examiner is requested to expedite the prosecution of this case by calling undersigned counsel for Applicant.

Respectfully submitted,
Dr. Bruce Horn
Applicant

Date: July 10, 2006

by: 
Jacques M. Dulin, Reg. No. 24,067
Attorney for Applicants

Jacques M. Dulin, Esq.
Innovation Law Group, Ltd.
237 North Sequim Avenue
Sequim, WA 98382-3456
e-mail: Dulin@InnovationLaw.com
Phone #: 360-681-7305
Fax #: 360-681-7315

End of Section 3, Remarks
End of Response to Office Action